Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application. The following listing provides the amended claims with deleted material crossed out and new material underlined to show the changes made.

Listing of Claims:

1. (Currently Amended) A method comprising:

for an event to be logged that has not yet been logged within an application:

creating an event object, said event object occupying a memory space that is independent of said application; and

logging within said event object ~~the~~ a start time, end time and information regarding the event;

wherein said creating and said logging are performed on a single computer on which said application executes.

2. (Previously Presented) A method according to claim 1 further comprising:

checking whether event logging has been turned on for the event.

3. (Previously Presented) A method according to claim 2, wherein said creating and said logging are performed for each event having event logging turned on, wherein a plurality of event objects are created and logged for a plurality of events.

4. (Previously Presented) A method according to claim 3 further comprising:

analyzing said event objects after event logging is turned off.

5. (Previously Presented) A method according to claim 4, wherein analyzing includes:

allowing user definition of the hierarchical levels of granularity of said events whose event objects are to be analyzed; and

allowing user definition of contexts for differentiating repeated occurrences of events deemed identical by nature of their hierarchical position.

6. (Previously Presented) A method according to claim 5, wherein analyzing further includes:

grouping events into their hierarchical subgroups; and

grouping events by their context, if any are defined.

7. (Previously Presented) A method according to claim 6, wherein analyzing comprises:

traversing through the hierarchy of subgroups until the subgroup of finest granularity is traversed;

subdividing said events into further subgroups;

computing statistics for each subgroup while traversing; and

displaying said statistics.

8. (Previously Presented) A method according to claim 7, wherein if said subgroup of finest granularity has been traversed, then:

aggregating events deemed identical by virtue of their hierarchical position into an aggregate;

computing statistics for each aggregate; and

displaying said statistics for each said aggregate.

9. (Previously Presented) A method according to claim 7, wherein said analyzing includes:

aggregating events deemed identical by virtue of their context into an aggregate;

computing statistics for each aggregate; and

displaying said statistics for each said aggregate.

10. (Currently Amended) A ~~system~~ computer comprising computer readable storage for storing:

_____a foundational layer upon which applications are built or executed; and

_____an event logging mechanism created by said foundational layer, said logging mechanism executing independently of said applications, said mechanism for identifying a set of events and for generating an event log for any of said applications, without referencing any event logs of said applications, ~~said logging mechanism for turning on or off at any time during the execution of said applications by an entity external to said applications~~ wherein each of said events is designated an enabled/disabled status, wherein a disabled status disables all logging for an event.

11. (Currently Amended) A ~~system~~ computer according to claim 10, wherein said event logging mechanism logs start time, end time and other event information into ~~the~~ an event object for each event to be logged.

12. (Currently Amended) A ~~system~~ computer according to claim 10, wherein said foundational layer is an operating system.

13. (Currently Amended) A ~~system~~ computer according to claim 10, wherein said foundational layer is a programmable framework.

14. (Currently Amended) A ~~system~~ computer according to claim ~~11~~10, wherein said event logging mechanism can be turned on and then off from beyond ~~the~~ an execution space of said applications within said foundational layer, said turning on and off separate for each event.

15. (Currently Amended) A ~~system~~ computer according to claim 10, wherein said event logging mechanism can be turned on and turned off and configured using a browser application running on said computer.

16. (Currently Amended) A ~~system~~ computer according to claim 15, wherein said event logging mechanism generates a plurality of event objects and is configured to analyze said event objects and present to said browser application the results thereof.

17. (Currently Amended) A ~~system~~ computer according to claim 16, wherein said event logging mechanism is configured to analyze said event objects based upon hierarchical and contextual grouping.

18. (Currently Amended) A ~~system~~ computer to claim 16, wherein said event logging mechanism is configured to aggregate said event objects deemed identical based upon at least one of hierarchical and contextual grouping.

19. (Currently Amended) An article comprising a computer readable medium ~~having instructions stored thereon~~ storing a computer program for execution by at least one processor, the computer program comprising a set of instructions which when executed causes:

for each event in a plurality of events to be logged that has not yet been logged within an application,

creating an event object, said event object occupying a memory space that is independent of said application; and

logging within said event object the start time, end time and information regarding the event;

wherein said creating and said logging are performed on a single computer on which said application executes.

20. (Currently Amended) An article according to claim 19, wherein the computer program further comprises a set of instructions ~~having further instructions stored thereon~~ which when executed causes:

analyzing of said event objects according to hierarchical and contextual grouping.

21. (Previously Presented) An apparatus comprising:

means for creating, for an event to be logged that has not yet been logged within an application, an event object, said event object occupying a memory space that is independent of said application; and

means for logging within said event object the start time, end time and information regarding the event;

wherein said creating and said logging are performed on a single computer on which said application executes.

22. (Previously Presented) An apparatus according to claim 21 further comprising:

means for analyzing of said event object according to hierarchical and contextual grouping.

23. (Currently Amended) A ~~system~~ computer comprising computer readable storage for storing:

_____a foundational layer upon which applications are executed;

_____a first application for executing on said foundational layer;

_____a second application for executing on said foundational layer; and

_____an event-logging mechanism for execution on said foundational layer, for functioning interoperably with but separately from said first and second applications, and for generating an event log for each of said first and second applications, wherein at least one of said first and second applications does not generate an event log, wherein said event-logging mechanism is separate from said first and second applications and is not compiled with said applications, wherein the event logging mechanism creates an event object for each of said events, each event object designated for log information to be stored and later accessed for analysis.

24. (Currently Amended) A ~~system~~ computer according to claim 23, wherein said generating an event log comprises storing, for each event to be logged, a temporal attribute of an event in ~~the~~ an event object associated with the event.

25. (Currently Amended) A ~~system~~ computer according to claim 23, wherein said event-logging mechanism comprises analyzing of said event log according to hierarchical and contextual grouping.

26. (Currently Amended) A ~~system~~ computer according to claim 23 further comprising a first area of memory allocated to the first application, a second area of memory allocated to the second application and a third area of memory allocated to the event logging mechanism, wherein said third area of memory is separate from the areas of memory allocated to the first and second applications.

27. (Currently Amended) A ~~system~~ computer according to claim 26 further comprising an enable/disable state for each event identified by the application wherein the disable state precludes any system from creating an event log.

28. (Currently Amended) A ~~system~~ computer according to claim 26, wherein generating an event log is performed for each event having event logging enabled.

29. (Currently Amended) ~~The system~~ A computer according to claim 23, wherein the foundational layer is an operating system upon which applications are executed.

30. (Previously Presented) An event logging method comprising:

for each of a plurality of events that need to be logged but have not yet been logged within a plurality of applications:

creating an event object;

storing said event object in a first memory space that is uniquely allocated for the event logging method, said first memory space separate from a second memory space allocated for the plurality of applications; and

logging within said event object the start time, end time and information regarding the event;

wherein said first and second memory spaces are within a third memory space of a single computer.

31. (Currently Amended) A method according to claim 30 further comprising creating, for the event ~~object~~, an enabled/disabled status wherein the disabled status disables all logging for the event within a system that includes the plurality of applications.

32. (Previously Presented) A method according to claim 30 further comprising checking, for each event identified by an application within the plurality of applications, whether event logging has been enabled.

33. (Previously Presented) A method according to claim 32 further comprising analyzing of said event objects after event logging is disabled.

34. (Previously Presented) A method according to claim 30, wherein the memory space occupied by the event log is within memory space that has been allocated solely to the event logging mechanism.

35. (Previously Presented) A method according to claim 30, wherein the events that are logged by the event logging mechanism have not been previously logged by any other application.

36. (Previously Presented) A method according to claim 30, wherein information placed in the event log is first logged by the event logging mechanism.

37. (Currently Amended) A method according to claim 30 further comprising an enable/disable state <u>for each event,</u> wherein the disable state precludes any system from creating an event log.

38. (Previously Presented) A method according to claim 30, wherein said creating is done by a foundational layer that is a development framework.

39. (New) A method comprising:

for an event to be logged that has not yet been logged within an application:

creating an event object, said event object occupying a memory space that is independent of said application, said event object designated for log information to be stored and later accessed for analysis; and

logging within said event object a start time, end time and information regarding the event;

wherein said creating and said logging are performed on a single computer on which said application executes.

40. (New) A method according to claim 39 further comprising:

checking whether event logging has been turned on for the event;

wherein said creating and said logging are performed for each event having event logging turned on, wherein a plurality of event objects are created and logged for a plurality of events.

41. (New) A method according to claim 40 further comprising:

analyzing said event objects after event logging is turned off.

42. (New) A computer comprising storage for:

a foundational layer upon which applications are executed; and

an event-logging mechanism for execution on said foundational layer, for functioning interoperably with but separately from said applications, said mechanism for

identifying a set of events and generating an event log for each of said applications, wherein at least one of said applications does not generate an event log, wherein the event logging mechanism creates an event object for each of said events, each event object designated for log information to be stored and later accessed for analysis.

43. (New) A computer according to claim 42, wherein said generating an event log comprises storing, for each event to be logged, a start time, end time and information regarding the event.

44. (New) A computer according to claim 42, wherein said event-logging mechanism comprises analyzing of said event log according to hierarchical and contextual grouping.

45. (New) A computer according to claim 44 further comprising an enable/disable state for each event identified by the application, wherein the disable state precludes the system from creating an event log, wherein generating an event log is performed for each event having event logging enabled.